

DESAIN DATA BASE

Desain Data Base

- Proses Desain Data Base
 - Analisis Persyaratan
 - Desain Data Base Konseptual
 - Desain Data Base Logika
 - Perbaiki Skema
 - Desain Data Base Fisik
 - Desain Aplikasi dan Keamanan

Analisis Persyaratan

- Memahami dan mengetahui data yang harus disimpan dalam database, aplikasi apa yang harus dibangun di atasnya, jenis operasi apa yang akan digunakan, dst.

Desain Database konseptual

- Informasi yang dikumpulkan pada tahap analisis persyaratan digunakan untuk mengembangkan deskripsi data tingkat tinggi.
- Tahapan ini sering dilakukan dengan menggunakan model E-R.
- Tujuannya adalah untuk menciptakan gambaran sederhana tentang data yang mirip dengan pemikiran pengguna / user.

Desain Database Logika

- Merubah skema E-R menjadi skema database relasional.
- Tujuannya adalah memperoleh skema konseptual pada model data relasional yang sering dinamakan skema logika.

Perbaikan Skema

- Analisis sekumpulan relasi dalam skema database relasional untuk mengidentifikasi permasalahan yang muncul dan memperbaikinya.

Desain Data Base Fisik

- Tahapan ini mencakup pembuatan indeks pada beberapa tabel, mengelompokkan beberapa tabel atau melibatkan desain ulang yang substansial terhadap beberapa bagian skema database.

Desain Aplikasi dan Keamanan

- Semua proyek perangkat lunak yang melibatkan sebuah DBMS harus mempertimbangkan aspek aplikasi yang berada di luar database.
- Enkripsi
- Digital Signature

Entity-Relationship (E-R)

- Model data Entity-Relationship (E-R) terdiri dari sekumpulan obyek-obyek, yang disebut dengan entitas dan hubungan yang terjadi diantara obyek-obyek tersebut.
- Model data E-R terbagi menjadi tiga konsep dasar, yaitu : himpunan entitas, himpunan relationship, dan atribut.

Entitas (*Entity*) dan Himpunan Entitas (*Entity Set*)

- Suatu entitas merupakan suatu obyek dasar atau individu yang mewakili sesuatu yang nyata eksistensinya dan dapat dibedakan dari obyek-obyek yang lain.
- Suatu entitas mempunyai sekumpulan sifat, dan nilai dari beberapa sifat tersebut adalah unik yang dapat mengidentifikasi entitas tersebut.

Entitas

- Sebagai contoh : setiap mahasiswa pada suatu perguruan tinggi merupakan suatu entitas, dan NIM secara unik dapat menjadi identitas seorang mahasiswa.

Himpunan Entitas

- Sekumpulan entitas yang mempunyai tipe yang sama (sejenis) dan berada dalam lingkup yang sama membentuk suatu himpunan entitas
- Sebagai contoh, sekumpulan orang yang menjadi mahasiswa pada suatu perguruan tinggi, dapat didefinisikan sebagai himpunan entitas *mahasiswa*. Sehingga dapat dikatakan bahwa entitas menunjuk pada individu suatu objek, sedangkan himpunan entitas menunjuk pada rumpun dari individu tersebut.
- Suatu entitas memiliki atribut. Atribut merupakan sifat-sifat atau properti yang dimiliki oleh entitas. Atribut inilah yang membedakan antara entitas yang satu dengan entitas yang lain.

- Sebagai contoh, entitas mahasiswa mempunyai atribut NIM, nama_mhs, alamat_mhs, tgl_lahir

NIM	Nama_mhs	Alamat_mhs	Tgl_lahir
0700001	Pascal	Menjangan 9 Subang 20236	25-2-1975
0700002	Rudi	Beruang 12 Jogjakarta 60342	10-5-1993
0600002	Firdaus	Mliwis 7 Semarang 50234	9-12-1998
0500003	Ruben	Durian Raya Semarang 50276	20-10-1977
0400034	Fajar	Rambutan 21 Liwa 80431	3-11-1968

Atribut dalam E-R

- Dikarakterisasikan dalam beberapa tipe :
 - Atribut sederhana dan komposit
 - Atribut bernilai tunggal dan banyak
 - Atribut null
 - Atribut turunan

Atribut Sederhana dan Komposit

- Atribut sederhana yaitu atribut yang tidak diuraikan menjadi sub atribut, atau tidak dapat diuraikan menjadi atribut yang lain.
- Atribut komposit adalah atribut yang dapat diuraikan menjadi beberapa sub atribut.
- Pada contoh di atas, nama_mhs merupakan atribut sederhana, sedangkan alamat_mhs merupakan atribut komposit karena dapat diuraikan menjadi beberapa sub atribut, yaitu alamat, nama_kota, kodepos.

Atribut bernilai tunggal dan banyak

- Atribut bernilai tunggal ditujukan pada atribut yang mempunyai paling banyak satu nilai untuk setiap baris data.
 - Contoh data mahasiswa di atas, semua atribut merupakan atribut bernilai tunggal.
- Atribut bernilai banyak adalah atribut yang dapat berisi lebih dari satu nilai, tetapi mempunyai jenis yang sama.
 - Contoh atribut bernilai banyak adalah jika kita menambahkan atribut hobi pada data mahasiswa. Seorang mahasiswa dapat memiliki satu hobi atau lebih dari satu hobi, bahkan dapat juga tidak mempunyai hobi.

Atribut Null

- Nilai null digunakan ketika suatu entitas tidak mempunyai suatu nilai untuk suatu atribut. Null juga menandakan bahwa suatu nilai atribut tidak ada atau tidak diketahui.
- Sebagai contoh, jika mahasiswa tidak mempunyai hobi, maka pada atribut hobi tidak ada nilainya.

Atribut Turunan

- Atribut turunan adalah atribut yang nilainya diperoleh dari pengolahan atau dapat diturunkan atribut atau tabel lain yang berhubungan.
- Sebagai contoh adalah penambahan atribut angkatan pada tabel data mahasiswa. Nilai pada atribut angkatan dapat diketahui dari NIM mahasiswa.

Relasi dan Himpunan Relasi

- Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari sejumlah himpunan entitas yang berbeda.
- Misalnya, seorang mahasiswa dengan NIM = '0700001' dan nama_mhs = 'Pascal' (yang ada di himpunan entitas mahasiswa) mempunyai relasi dengan entitas mata kuliah dengan kode_mtkul = 'CS3004' dan nama_mtkul = 'Analisa Algoritma'. Relasi dari kedua entitas tersebut mengandung arti bahwa mahasiswa tersebut sedang mengambil mata kuliah Analisa Algoritma.

Kode_mtkul	Nama_mtkul	sks	Semester
CS3004	Analisa Algoritma	3	3
CS4005	Pengolahan Citra	3	4
CS5006	Fuzzy Logic	3	5
CS5008	Genetic Algorithm	3	5

Kardinalitas Relasi

- Kardinalitas merupakan jumlah maksimum entitas, dimana entitas tersebut dapat berelasi dengan entitas pada himpunan entitas yang lain.

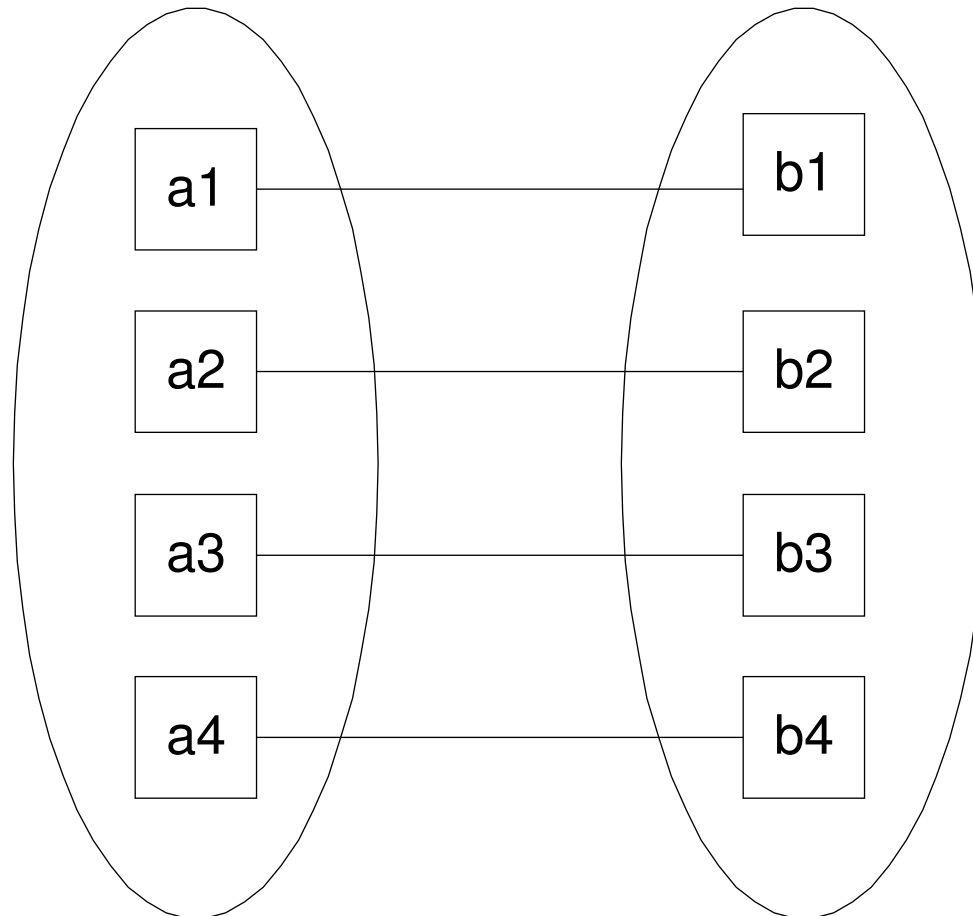
Tipe – tipe Kardinalitas Relasi

- Satu – satu (One to One)
- Satu – Banyak (One to Many)
- Banyak – Satu (Many to One)
- Banyak – Banyak (Many to Many)

Satu – satu (One to One)

- Suatu entitas di dalam himpunan entitas A dihubungkan dengan paling banyak satu entitas di dalam himpunan entitas B, dan entitas di dalam himpunan entitas B dihubungkan dengan paling banyak satu entitas dalam himpunan entitas A.

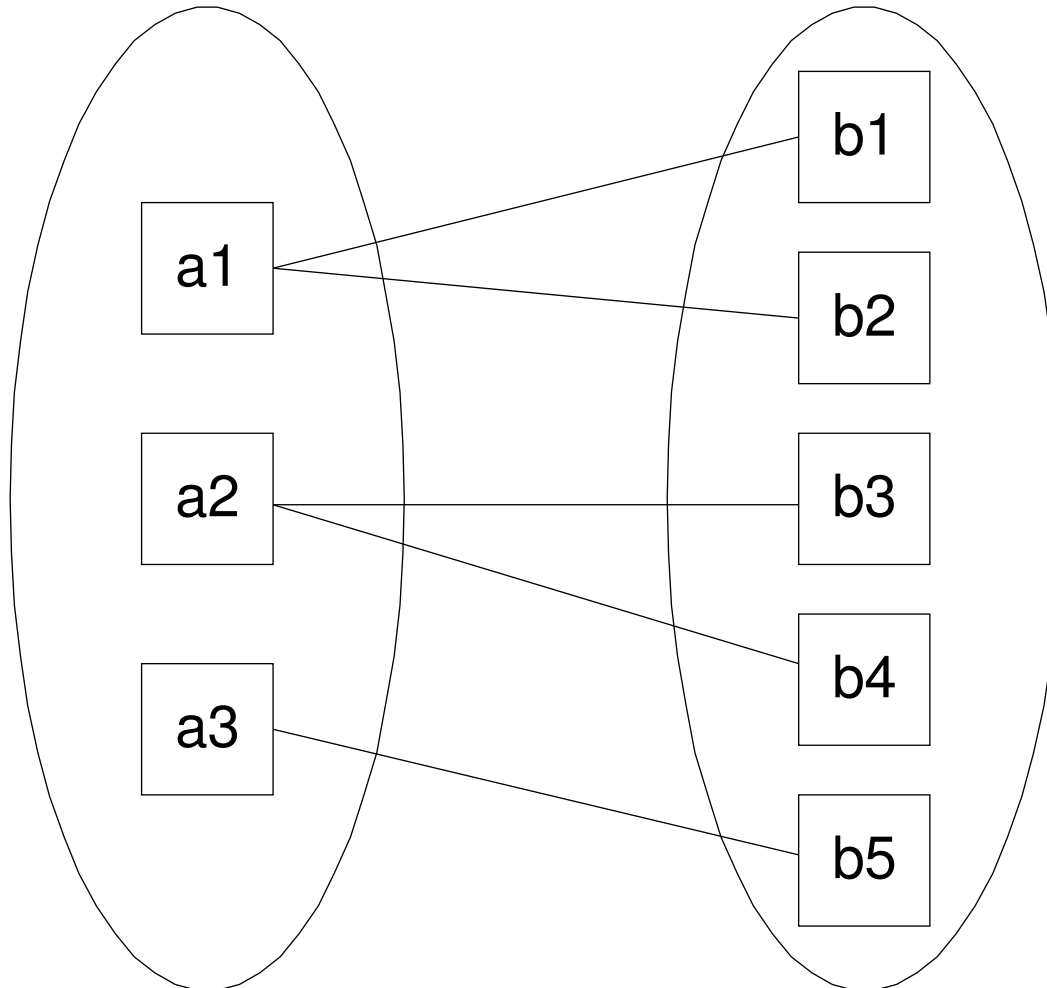
Satu – satu (One to One)



Satu – Banyak (One to Many)

- Suatu entitas di dalam himpunan entitas A dihubungkan dengan lebih dari satu entitas di dalam himpunan entitas B, dan entitas di dalam himpunan entitas B hanya dapat dihubungkan dengan paling banyak satu entitas dalam himpunan entitas A.

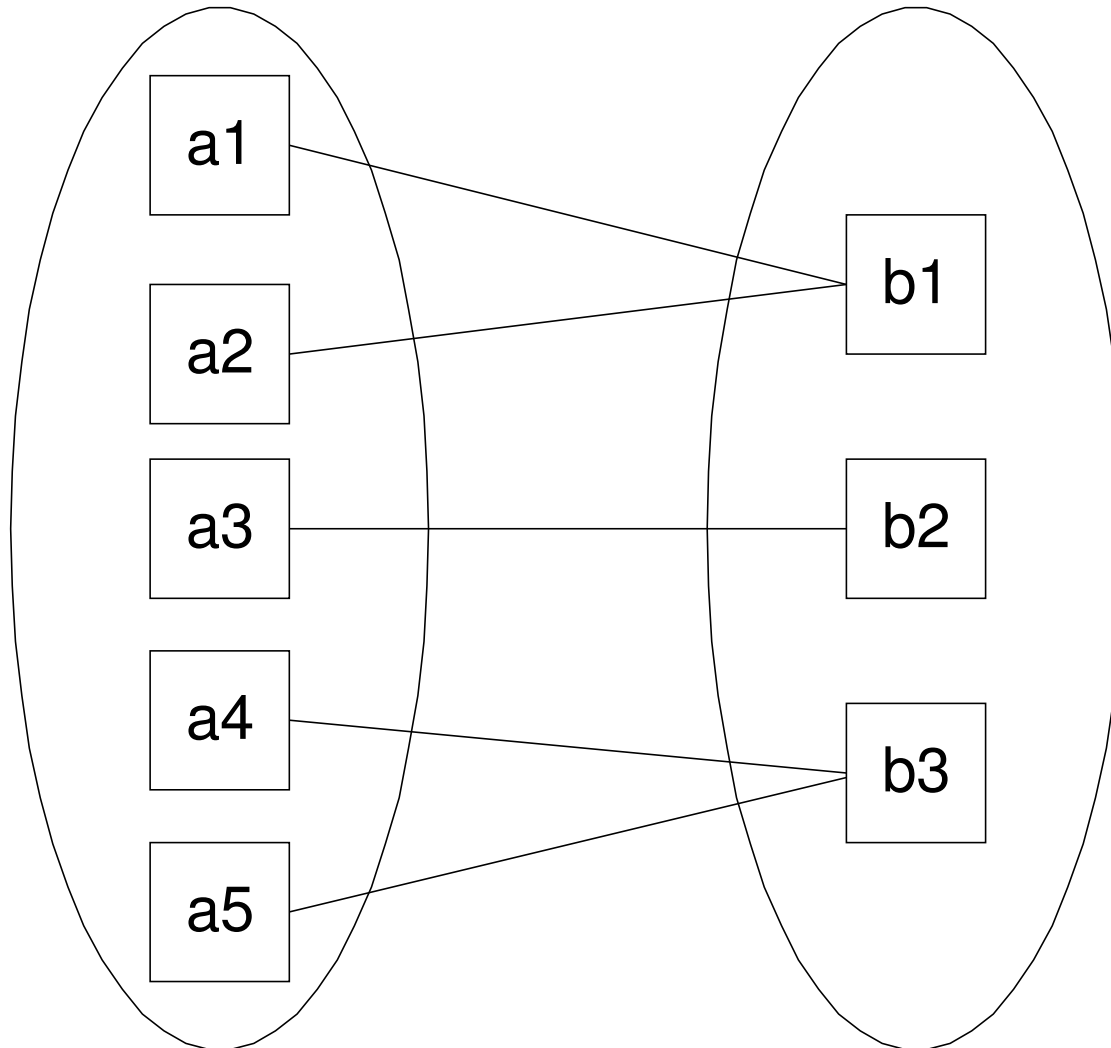
Satu – Banyak (One to Many)



Banyak – Satu (Many to One)

- Suatu entitas di dalam himpunan entitas A dihubungkan dengan paling banyak satu entitas di dalam himpunan entitas B, dan entitas di dalam himpunan entitas B dapat dihubungkan dengan lebih dari satu entitas dalam himpunan entitas A.

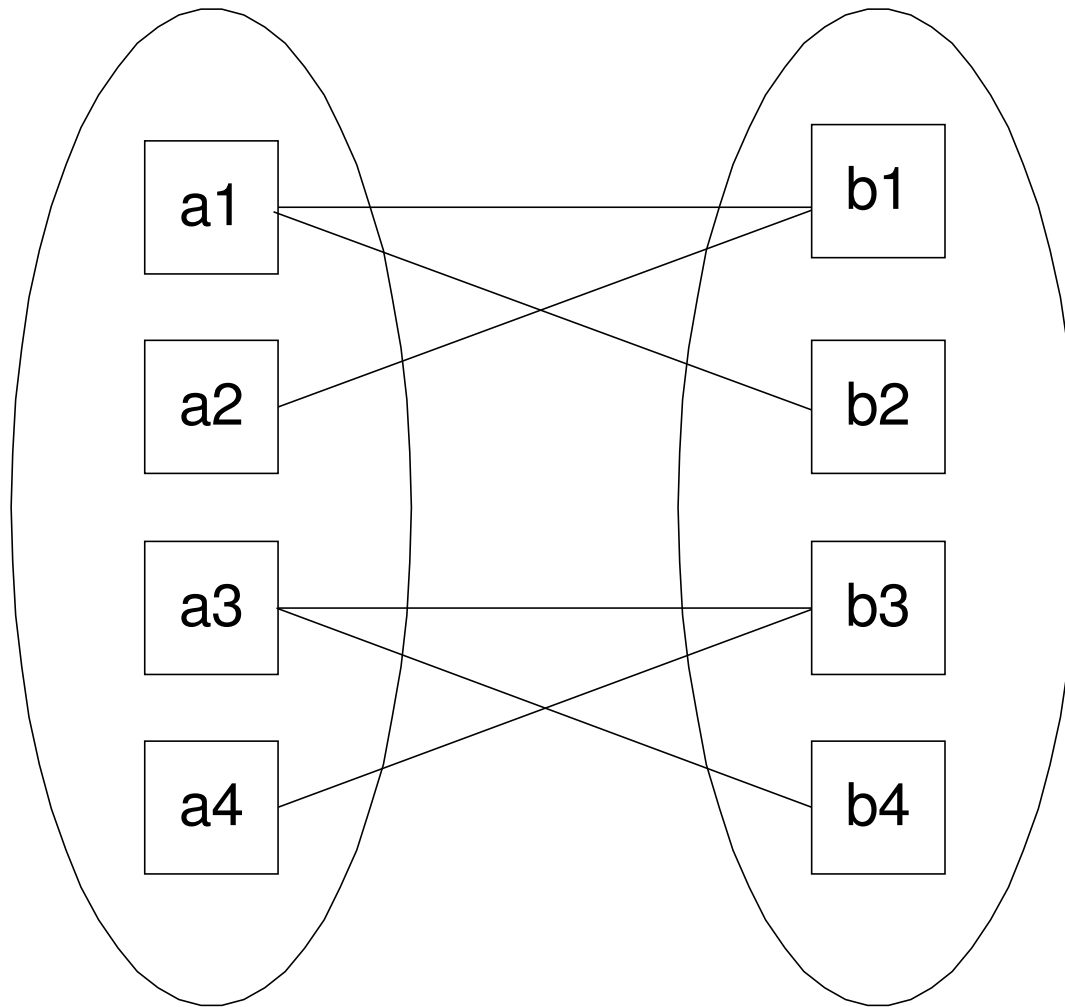
Banyak – Satu (Many to One)



Banyak-banyak (Many to Many)

- Suatu entitas di dalam himpunan entitas A dapat dihubungkan dengan lebih dari satu entitas di dalam himpunan entitas B, dan entitas di dalam himpunan entitas B dapat dihubungkan dengan lebih dari satu entitas dalam himpunan entitas A.

Banyak-banyak (Many to Many)



Kunci (Key)

- Kunci atau *key* merupakan satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data dengan unik.
- Jika suatu atribut menjadi kunci, maka tidak boleh ada dua atau lebih baris data dengan nilai yang sama.

Kunci (Key)

- Super Key
- Candidate Key
- Primary Key

Super Key

- Superkey, merupakan suatu himpunan yang terdiri dari satu atau lebih atribut yang dapat membedakan setiap baris data dengan unik dalam sebuah tabel.
- Sebagai contoh (NIM, nama_mhs, alamat_mhs, tgl_lahir), (NIM, nama_mhs, alamat_mhs), (NIM, nama_mhs), (nama_mhs), (NIM).

Candidate Key

- Candidate-key, merupakan himpunan atribut minimal yang dapat membedakan setiap baris data dengan unik dalam sebuah tabel.
- Sebagai contoh adalah (NIM), (nama_mhs).

Primary Key

- Primary key, merupakan kunci yang paling unik, lebih ringkas, lebih sering dijadikan acuan.
- Primary key dapat dipilih dari salah satu dari candidate key.
- Contoh primary key adalah (NIM).

Diagram Entity-Relationship (Diagram E-R)

- Suatu database, dapat digambarkan secara grafik dengan suatu diagram E-R.

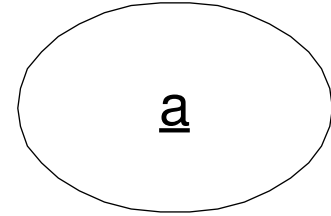
Notasi E-R

- Persegi panjang, menggambarkan himpunan entitas.
- Elips, menggambarkan atribut-atribut (atribut yang berfungsi sebagai kunci digarisbawahi).
- Belah ketupat, menggambarkan himpunan relationship.
- Garis, menggambarkan hubungan atribut ke entitas dan himpunan entitas ke himpunan relationship.
- Kardinalitas relasi dapat digambarkan dengan banyaknya garis cabang atau dengan pemakaian angka, 1 untuk relasi satu, n untuk relasi banyak.

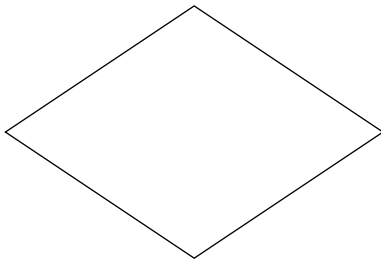
Notasi E-R



Himpunan Entitas E



atribut a sebagai key



Himpunan Relasi R



garis

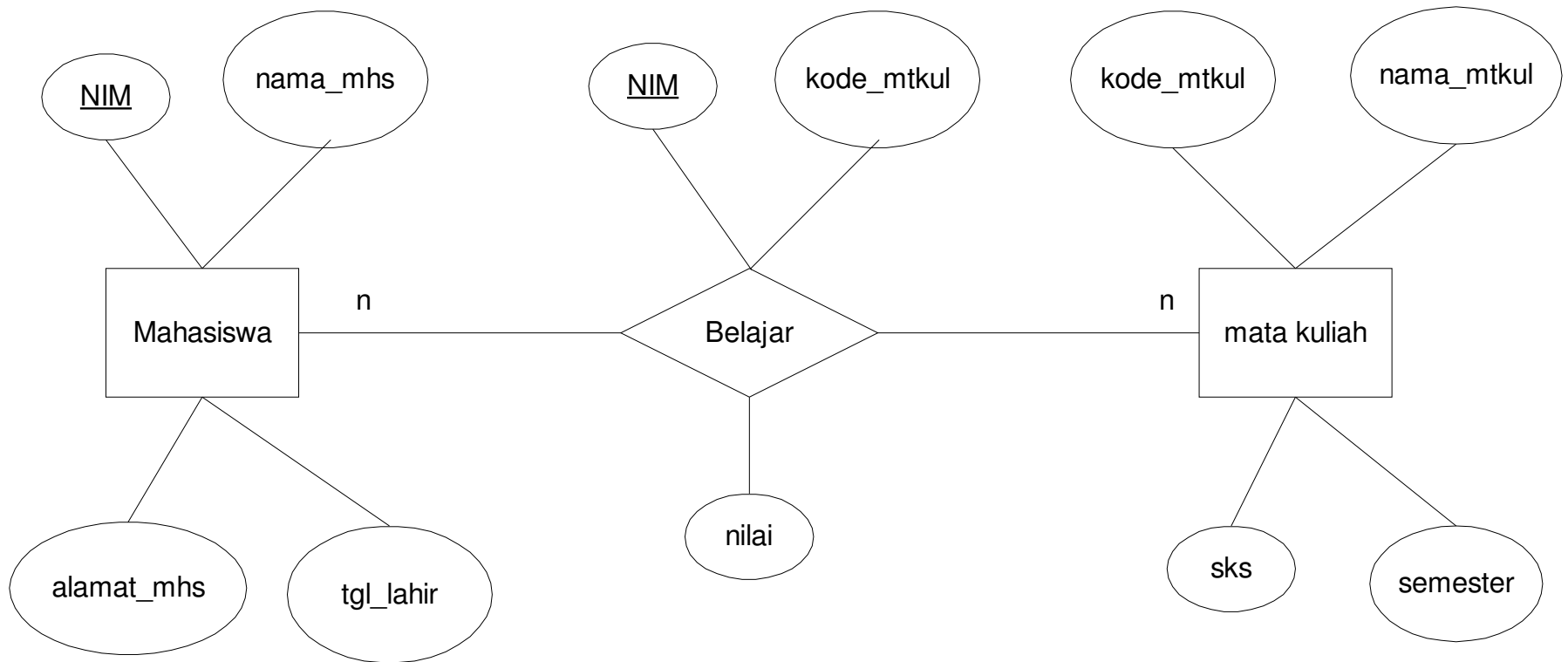
Tahapan Pembuatan Diagram E-R

- Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat
- Menentukan atribut – atribut key dari masing-masing himpunan entitas
- Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas yang ada beserta foreign key
- Menentukan derajat kardinalitas untuk setiap himpunan relasi
- Melengkapi himpunan entitas dan himpunan relasi dengan atribut deskriptif (non key)

Diagram E-R

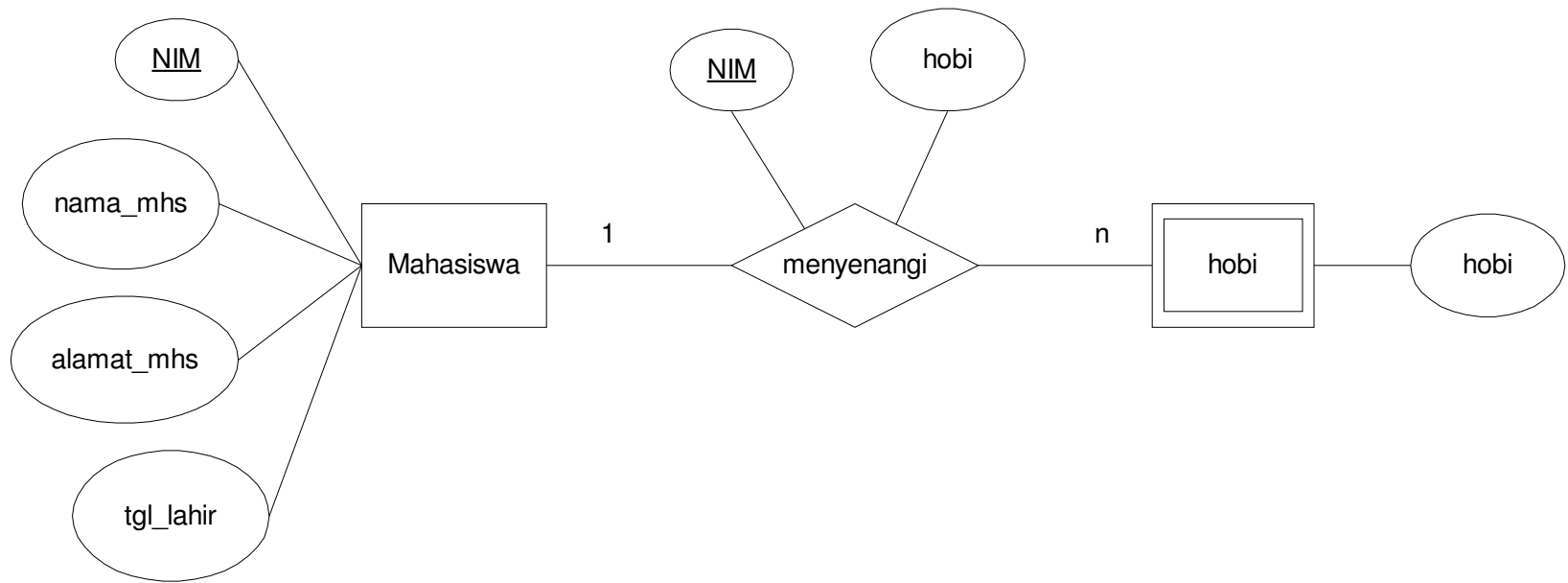
- Contoh, relasi antara himpunan entitas Mahasiswa dengan himpunan Mata Kuliah. Himpunan relasinya disebut dengan “Belajar”.
- Pada relasi ini, seorang mahasiswa bisa mengambil lebih dari satu mata kuliah. Setiap mata kuliah dapat diambil lebih dari satu mahasiswa.

Diagram E-R



Himpunan Entitas Lemah (Weak Entity Set)

- Himpunan entitas yang tidak memiliki atribut yang dapat berfungsi sebagai primary key.
- Himpunan entitas lemah berisi entitas-entitas yang kemunculannya tergantung pada keberadaannya dalam sebuah relasi terhadap entitas lain (entitas kuat).
- Sebagai contoh, pada tabel data Mahasiswa, dapat dilengkapi dengan entitas baru, yaitu hobi.

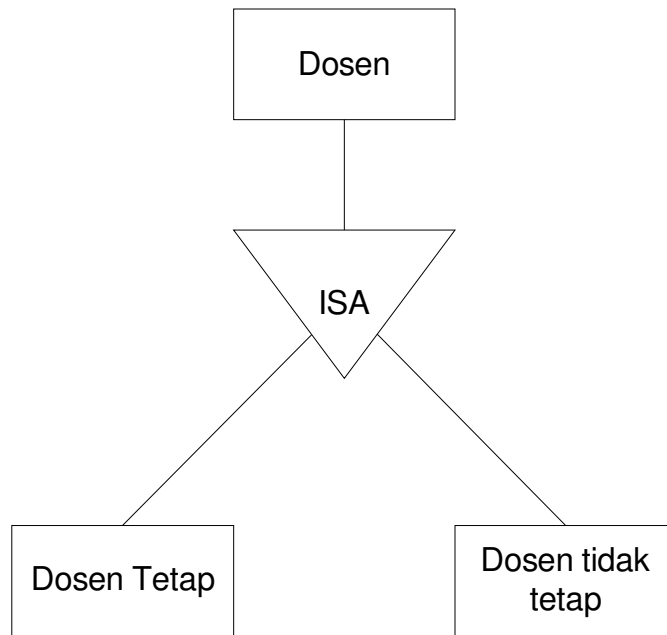


Data hobi dapat dikategorikan sebagai himpunan entitas lemah (digambarkan dengan kotak ganda), karena keberadaannya bergantung pada adanya relasi dengan entitas yang ada pada himpunan entitas mahasiswa

Spesialisasi dan Generalisasi

- Suatu himpunan entitas mungkin mempunyai sub kelompok entitas yang berbeda dari entitas yang lain pada himpunan entitas tersebut.
- Suatu sub himpunan yang berada dalam himpunan entitas mungkin mempunyai atribut yang tidak dimiliki oleh semua entitas dalam himpunan entitas tersebut.

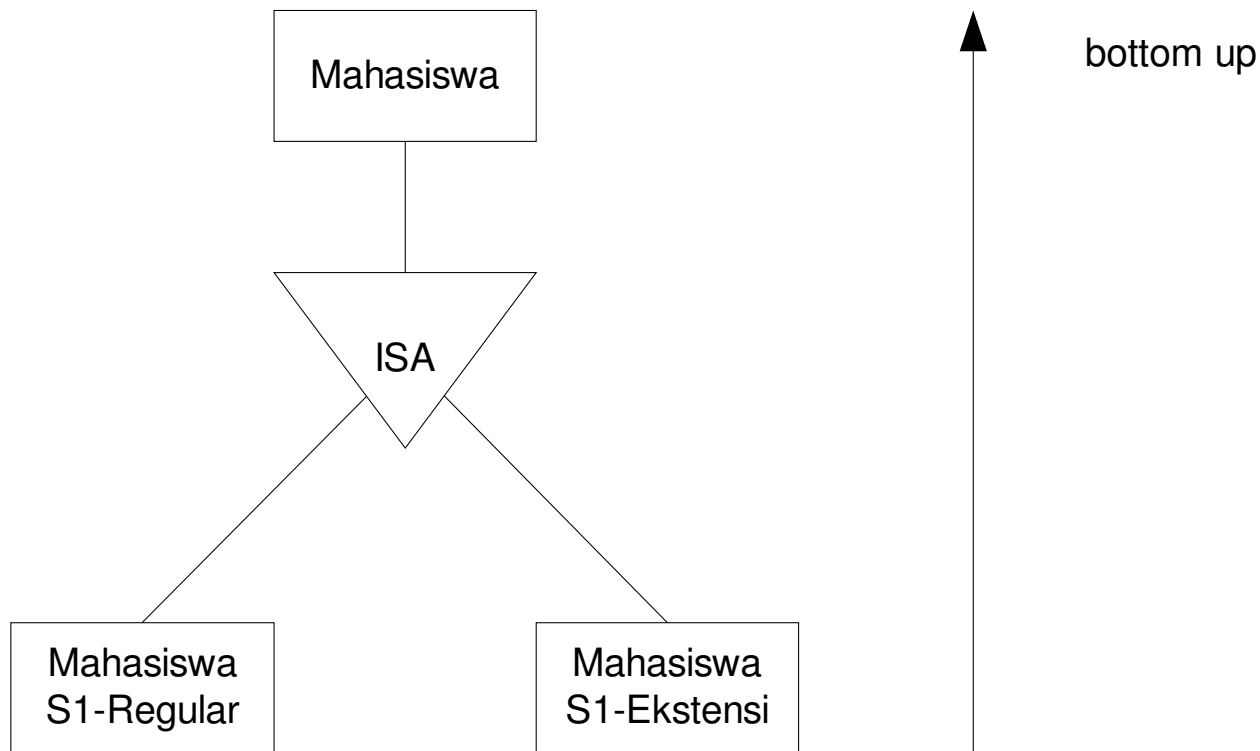
- Himpunan entitas dosen dapat dibagi menjadi dua, yaitu dosen tetap dan dosen tidak tetap. Himpunan entitas dosen, bisa memiliki atribut nip, golongan, pangkat, tgl_masuk.
- Sedangkan untuk dosen tidak tetap, atribut-atribut tersebut tidak diperlukan.
- Adanya perbedaan tersebut menyebabkan entitas dosen tidak mungkin disatukan dalam sebuah himpunan entitas.
- Dari suatu himpunan entitas, dapat dilakukan pengelompokan yang dapat membentuk suatu himpunan entitas baru atau proses top down disebut dengan proses **spesialisasi**



top down



Kebalikan dari spesialisasi adalah **generalisasi**,
yaitu proses bottom up.



Agregasi

- Satu keterbatasan dari model E-R adalah tidak mungkin untuk mengekspresikan suatu himpunan relasi yang secara langsung menghubungkan sebuah himpunan entitas dengan sebuah himpunan relasi. Untuk mengatasi hal tersebut, digunakan suatu notasi khusus yang dinamakan dengan agregasi.

Agregasi

- Contoh, relasi antara himpunan entitas mahasiswa dengan himpunan entitas mata kuliah. Terdapat beberapa mata kuliah yang mengandung kegiatan praktikum. Himpunan relasi “Kegiatan” dalam diagram ER berikut ini menunjukkan entitas mahasiswa yang mengikuti kegiatan praktikum, karena sedang mengambil mata kuliah yang ada praktikumnya

Agregasi

